EAX
A two-pass authenticated encryption mode

Mihir Bellare — U.C. San Diego

Phillip Rogaway — U.C. Davis and Chiang Mai University (Thailand)

David Wagner — U.C. Berkeley

Summary of our work

- "Authenticated encryption" (AE) modes of operation
  - Encrypt for confidentiality
  - Authenticate for integrity
- Goal: "Auth. encryption with associated data" (AEAD)
  - Support "associated data" (AD) - e.g., packet headers - that should be authenticated but not encrypted
- Additional goals:
  - Flexible, general-purpose, suitable for standardization
  - Patent-unencumbered
  - Provably secure
- Our solution: EAX

1st generation: ad-hoc schemes

- Many schemes proposed and used in practice:
  - CBC with xor checksum
  - PCBC
  - Kerberos: CBC with CRC checksum
  - IPSec's old ESP o AH
  - IPSec's new ESP
  - SSL/TLS
  - SSH
  - IEEE 802.11 WEP
  - IAPCBC

All of these have security defects!

- None of these were proven secure

2nd generation: provable security

- Generic-composition: encrypt-then-authenticate

- Advantages:
  + Provably secure [Bellare,Namprempre] [Krawczyk]
  + Supports associated data: a AEAD scheme
  + Unpatented

- Disadvantages:
  - Strict IV requirements if one uses standard enc schemes
  - More key material, longer key-setup time
  - No standard, no specs

3rd generation: One-pass provably secure AE(AD)

- IAPM [Jutla], OCB [Rogaway], XCBC [Gligor, Donescu]
- Advantages:
  + Encrypt and authenticate in one pass
  + Fast: takes about *n* block-cipher calls to process *n* blocks of data
- Disadvantages:
  - Some modes can't handle "associated data"
  - Some modes are not fully specified
  - All are patent-encumbered
- Due to patent concerns, adoption of these modes has been limited

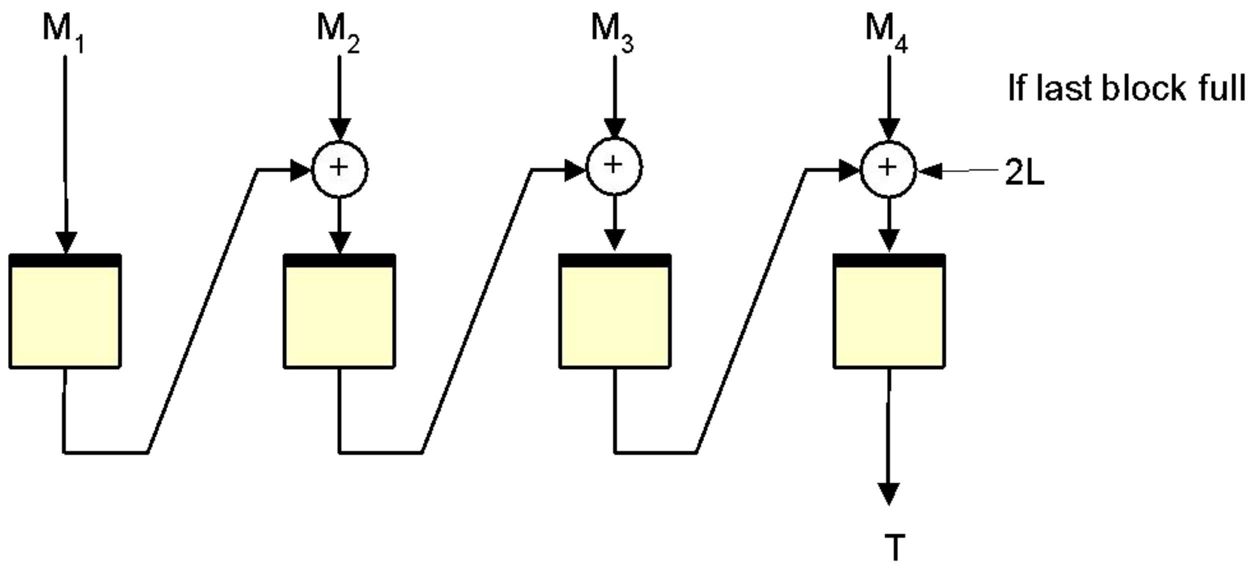
4th generation: Unpatented two-pass AEAD

- CCM: CTR + CBC-MAC [Whiting, Housley, Ferguson]
- EAX: builds on CTR and OMAC
- CWC: builds on CTR and hash127 [Kohno, Viega, Whiting]
- GCM: builds on CTR and GF(2^{128}) univ hash [Viega, Whiting]
- Caveat: Two-pass modes are typically ~ 2x slower than one-pass modes, in software

Comparison of 4th generation schemes

	CCM	EAX	CWC	GCM
Provably secure?	✔	✔	✔	✔
Unpatented?	✔	✔	✔	✔
Any length nonce?	□	✔	□	✔
One key?	✔	✔	✔	✔
On-line?	□	✔	✔	✔
Can preprocess static headers/AD?	□	✔	✔	✔
Fully parallelizable?	□	□	✔	✔
Preserves alignment?	□	✔	✔	✔
Fully specified?	✔	✔	✔	✔

[Iwata, Kurosawa]

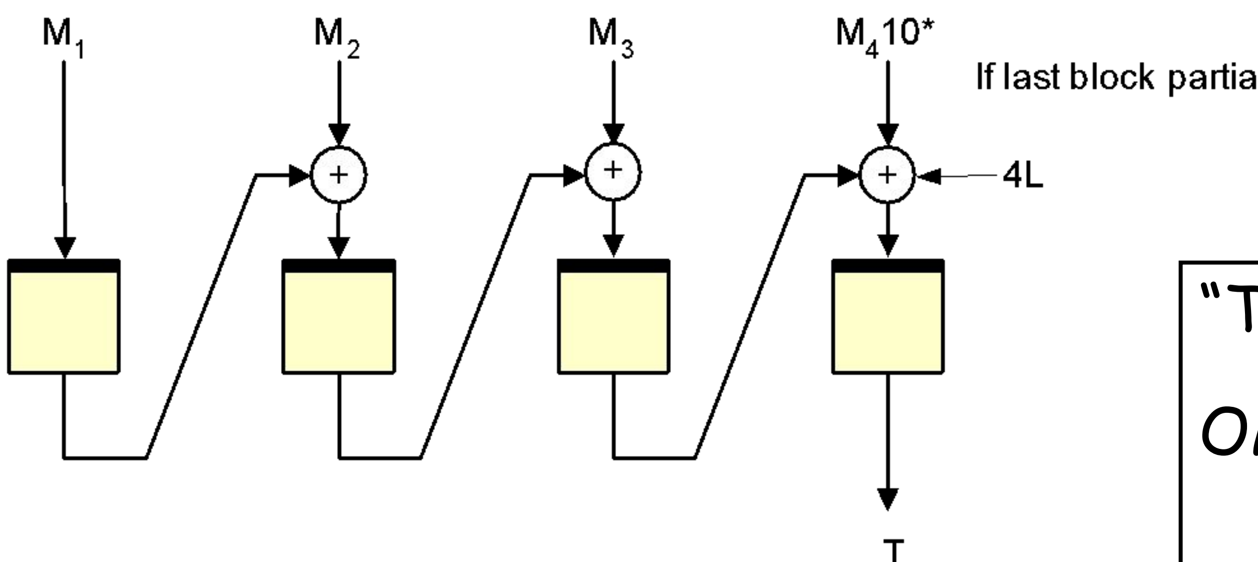


OMAC

$L = \pi(0^n)$

$2L = \text{msb}(L)?\ L{<<}1 :$
$\qquad L{<<}1 \oplus 0x87$

$4L = 2(2L)$



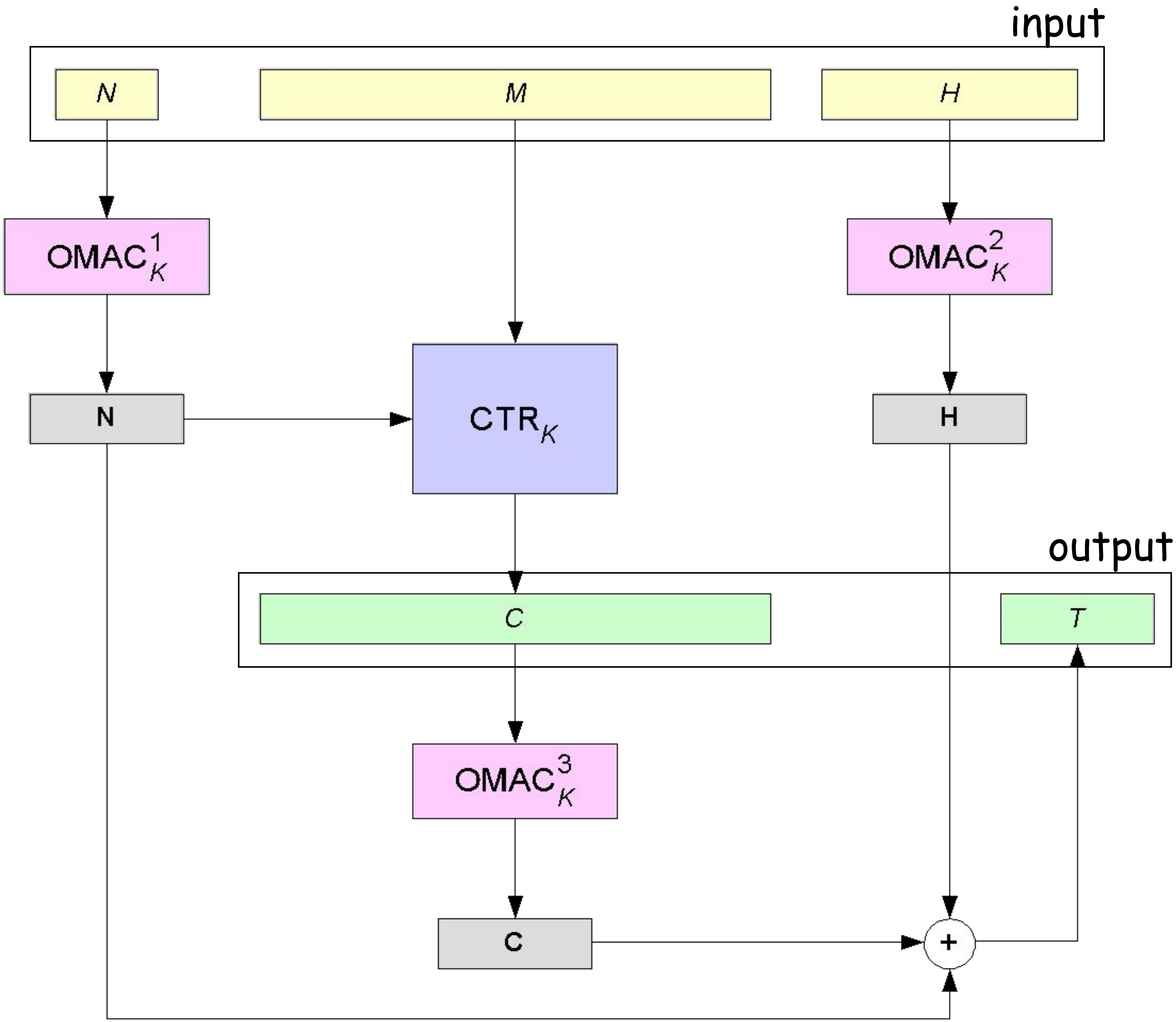
"Tweaked" OMAC:

$OMAC_k^T(x) =$
$\quad OMAC_k(T \,||\, x)$

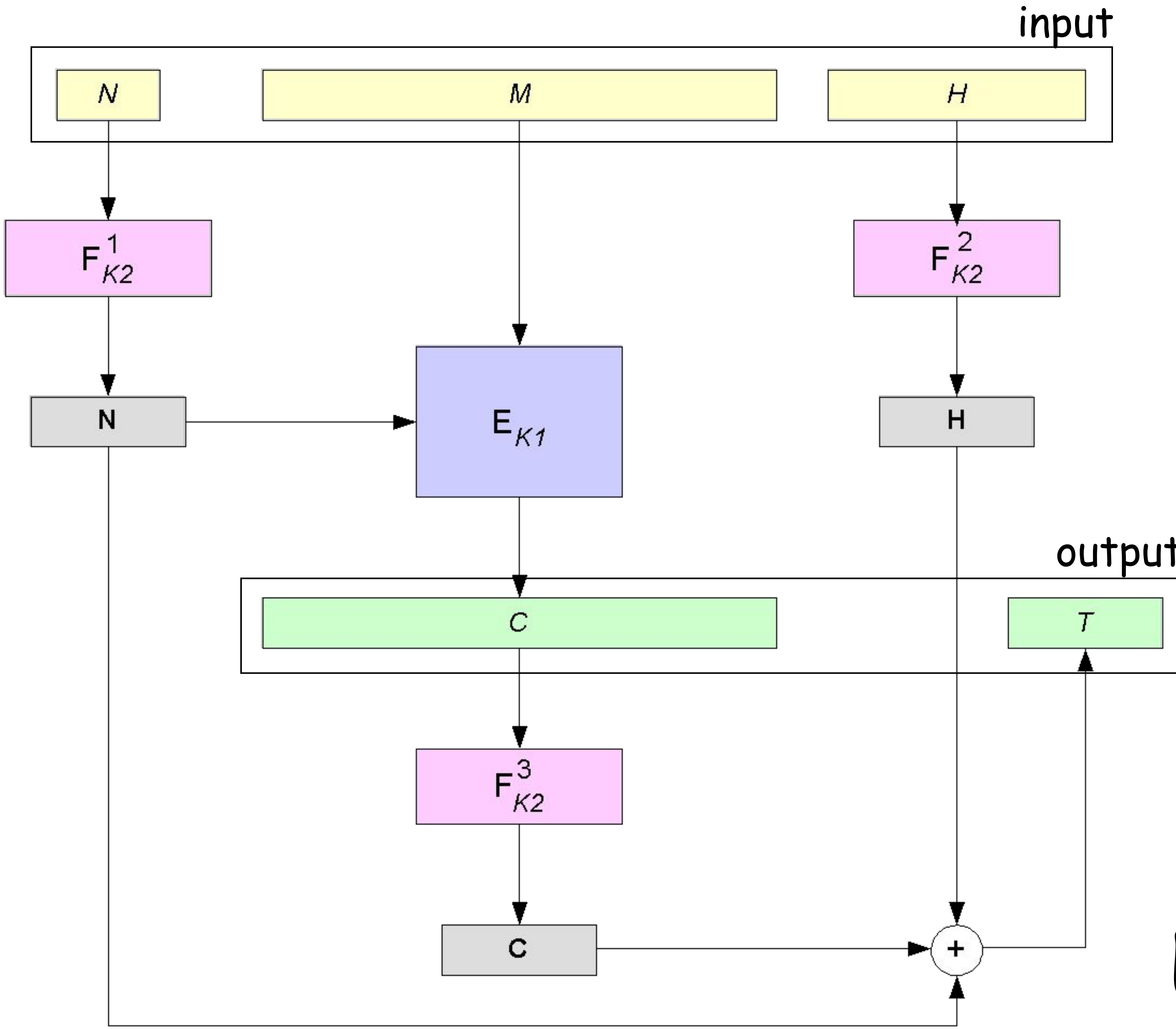
Security of OMAC'

Theorem [slight improvement of [IK]]
Suppose there is an adversary A that attacks $OMAC'[E]$ using time t and σ blocks worth of queries getting PRF-advantage $Adv^{prf}_{OMAC'[E]} = \delta$

Then there is an adversary B that attacks E using time t + tiny and $\sigma + 1$ blocks of text and getting PRP-advantage $Adv^{prp}_{E} = \delta - (\sigma+3)^2/2^n$



EAX



EAX2

Auth Encryption with Associated Data (AEAD)

Syntax of an AEAD scheme:

E: Key × Nonce × Header × Plaintext → Ciphertext
D: Key × Nonce × Header × Ciphertext → Plaintext ∪ {invalid}

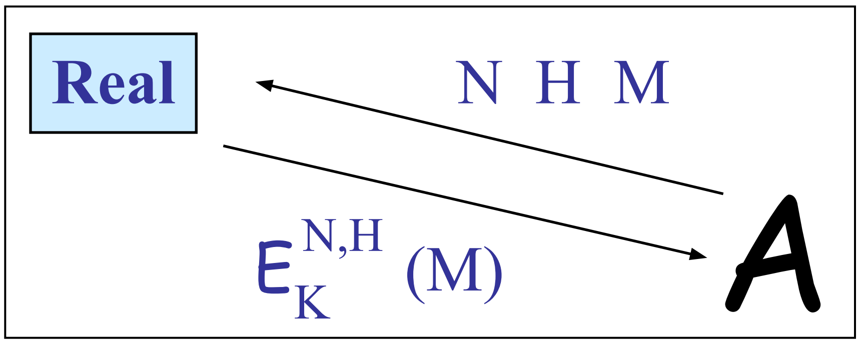
Security of an AEAD scheme:

- Privacy (≈ IND-CPA) next slide
- Integrity (≈ INT-CTXT) following slide

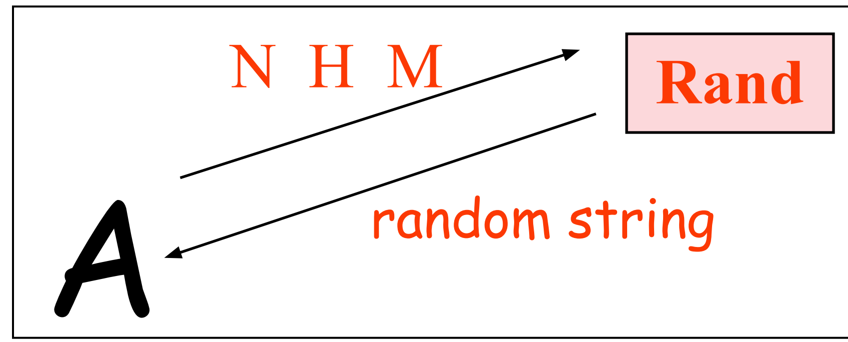
[RBB],[BDJR],[GM],[R]

Privacy of an AEAD Scheme

Real world



Ideal world



$$\mathbf{Adv}^{\mathrm{PRIV}}_{\Pi}(A) = \Pr[A^{\mathrm{Real}} = 1] - \Pr[A^{\mathrm{Rand}} = 1]$$

A is not allowed to repeat an N-value (nonces should be unique)

Integrity of an AEAD Scheme

Adversary *A* **forges** if it outputs N* H* C* s.t.

- C* is **valid** (it decrypts to a message, not to **invalid**)
- There was no earlier query N* H* M* that returned C*

Real

N H M

$E_K^{N,H}(M)$

A

N* H* C*

$$\mathbf{Adv}_{\Pi}^{\mathrm{AUTH}}(A) = \Pr[A^{\mathrm{Real}} \text{ forges}]$$

A is not allowed to repeat an N-value

Security of EAX

Theorem
Suppose there is an adversary A that attacks $EAX[E]$ using time t and σ blocks of chosen text getting privacy or authenticity $Adv_{EAX[E]} = \delta$.
Then there is an adversary B that attacks E using time t + tiny and σ + tiny blocks of text and getting PRP-advantage $Adv^{prp}_{E} = \delta - 11\sigma^2/2^n$.

If you believe that E is a good block cipher, you are forced to believe that EAX[E] is a good AEAD scheme.

Why use EAX?

- EAX is secure
  - Provably secure, if underlying block cipher is secure
  - Single API for naïve programmers avoids many pitfalls (e.g., poor IV handling, encrypt without auth, etc.)
- EAX is easy to use
  - One mode of operation provides everything you need
  - Nonces need only be non-repeating (don't need to be random)
  - Nonces, headers, and messages can be of any bit length
- EAX is good for performance
  - On-line: Can process streaming data on-the-fly
  - Can pre-process static headers
  - No encodings, no unaligned operations
  - Single key minimizes space and key-schedule operations
  - Caveat: EAX is 2x slower than IAPM/OCB/XCBC
- EAX is unpatented & free for all uses (as far as we know)

Questions?